GM

CRYPTOGRAPHIC INDUSTRY STANDARD

OF THE PEOPLE'S REPUBLIC OF CHINA

ICS 35.030

L 80

GM/T 0118-2022

Browser digital certificate application interface specification

浏览器数字证书应用接口规范

Issued on: November 20, 2022 **Implemented on: June 01, 2023**

Issued by: State Cryptography Administration

Table of Contents

Browser digital certificate application interface specification

1 Scope

This document specifies the browser SM2 digital certificate application interface; describes the definition of functions, data types and parameters of the digital certificate application interface in browsers, that support the application of domestic cryptographic algorithms.

This document is applicable to the development, application, testing of browser products, the development of browser applications that support SM2 digital certificates, the testing of secure browser password modules; it can also be used to guide the integration and development of third-party applications calling SM2 digital certificates in different terminal devices.

2 Normative references

The contents of the following documents constitute essential clauses of this document through normative references in the text. Among them, for dated references, only the version corresponding to that date applies to this document; for undated references, the latest version (including all amendments) applies to this document.

GB/T 20518 Information security technology - Public key infrastructure - Digital certificate format

GB/T 32918.2 Information security technology - Public key cryptographic algorithm SM2 based on elliptic curves - Part 2: Digital signature algorithm

GB/T 33560 Information security technology - Cryptographic application identifier criterion specification

GB/T 35275 Information security technology - SM2 cryptographic algorithm encrypted signature message syntax specification

GM/T 0016 Smart password key password application interface specification

GM/T 0100 Cryptographic application technical requirements for manually confirmed signing

GM/Z 4001 Cryptographic terminology

3 Terms and definitions

The terms and definitions as defined in GM/Z 4001, as well as the following terms and definitions, apply to this document.

3.1

SM2 cryptographic algorithm

Public key cryptographic algorithm defined by GB/T 32918.5.

3.2

Digital certificate

Also known as public key certificate, which is a data structure signed by a certificate authority (CA) that contains public key owner information, public key, issuer information, validity period, extended information. It can be divided into personal certificate, agency certificate, device certificate by type; or it can be divided into signature certificate and encryption certificate by purpose.

3.3

Digital signature

The result obtained by the signer using the private key to perform cryptographic operations on the signed data. The result can only be verified by the signer's public key and is used to confirm the integrity of the data to be signed, the authenticity of the signer's identity, the non-repudiation of the signature behavior.

3.4

Certificate context

A data structure, which is used to store relevant certificate information, including owner information, public key, issuer information, validity period, extended information.

3.5

Certificate storage

A logical concept, which is used to centrally store digital certificates, including different types of certificates such as user certificates and root certificates. The specific location of the certificate storage area is not defined and is implemented by the manufacturer implementing this interface.

3.6

Certificate store context

A data structure, which is used to store the certificate context and the SKF information corresponding to the certificate. The SKF information includes the SKF interface name, device name, application name, container name.

Note: SKF is the interface specification defined by GM/T 0016.

4 Abbreviations

The following abbreviations apply to this document.

AIA: Authority Information Access

API: Application Programming Interface

CRL: Certificate Revocation List

DER: Distinguished Encoding Rules

LDAP: Lightweight Directory Access Protocol

OCSP: Online Certificate Status Protocol

PEM: Privacy Enhancement Message

PIN: Personal Identification Number

5 Overall technical framework

The overall technical framework of the browser digital certificate application interface is shown in Figure 1.

Function description: Release the SKF link list.

Parameter:	pNodeHeader	[IN] Pointer to the SKF link list.
Return value	ERR_SSF_OK:	Success.
	Others:	Error code.

7.5 Relationship with other interface specifications

For GM/T 0019, its main purpose is to provide general cryptographic services such as certificate parsing, certificate authentication, confidentiality, integrity and non-repudiation of information to the cryptographic service layer and application layer, through a unified cryptographic service interface; convert the upper-layer cryptographic service request into a specific basic cryptographic operation request; call the corresponding cryptographic device through a unified cryptographic device application interface, to implement specific cryptographic operations and key operations.

For GM/T 0020, it is positioned between the application system and the typical cryptographic service interface, directly providing the application layer with certificate information parsing, digital certificate-based identity authentication, information confidentiality, integrity, non-repudiation and other advanced cryptographic services; redirecting the cryptographic service requests of the application system to the general cryptographic service interface; calling the corresponding cryptographic device through the general cryptographic service interface, to implement specific cryptographic operations and key operations.

For this specification, it does not care about specific cryptographic operations and key operations, but only enables browsers and third-party applications to quickly find SM2 digital certificates on the terminal operating system and obtain SKF interface information related to SM2 digital certificates.

In the above function definitions, some functions refer to GM/T 0019. The main difference is that the interface parameters in GM/T 0019 need to specify the hAppHandle handle, whilst the handle does not need to be specified in this document; for the SSF_GetCertStateByOCSP and SSF_GetCertStateByCRL interfaces in this document, the puiState parameter is used to return the obtained status, whilst the status is obtained through the function return value in GM/T 0019. The function list based on GM/T 0019 is shown in Table 17.

Appendix B

(Informative)

Routine using this specification interface

B.1 Register SKF and store certificates

This routine implements certificate registration. The SSF interfaces used in this routine are: SSF_RegisterSKF, SSF_AddCert.

The implementation process of this routine is:

a) Register SKF library;

b) Read SKF and the certificate read by the SKF interface;

c) Store certificate.

```
#define STORE_TYPE_USER 2

Int testdemo ()

{

    UINT32 uiRet = 0;

    SSF_Data stName = {strlen("SKFName"), (UINT8*)"SKFName"}; //Name

    SSF_Data stPath = {strlen("c:\\SKFName.dll"), (UINT8*)"c:\\SKFName.dll"};
    //Path

    //Linux and domestic operating system SKF path is usr/lib/libSKFName.so

    SSF_SKF pSKF = {stName, stPath, 0, 0};

    SSF_SKFCertRef       *pSKFCertRef = NULL;

    SSF_StoreContext_NODE * header= NULL;

    SSF_CertificateContext     * pCertCtx = NULL;

    SSF_CertificateFindAttr    *pCertificateFindAttr = NULL;


    //----------------------1. Register the certificate in the device to the system----------
    ---------------
```

```
#else
    dlclose(ghInst)
#endif
}
```

B.4 Certificate usage

This routine simulates the process of finding certificates stored in the certificate storage area. The SSF functions used in this routine are: SSF_EnumCert, SSF_FindCert, SSF_FreeStoreCertCtxLink.

The SKF interfaces used in this routine are: SKF_VerifyPIN, SKF_GetContainerType, SKF_ExportPublicKey, SKF_DigestInit, SKF_Digest, SKF_ECCSignData, SKF_RSASignData.

The implementation process of this routine is:

a) Enumerate (find) certificates;

b) Use certificates;

c) Release the certificate storage context list.

The sample code is as follows:

```
int testdemo2(char * pszPIN, ULONG * puiRetryCount)
{
    UINT32 uiRet = 0;
    SSF_StoreContext_NODE   *header = NULL;
    SSF_CertificateFindAttr      * pCertificateFindAttr = NULL;
    UINT32 ulContainerType; //Container type
    HAPPLICATION hAPP = NULL; //Application handle
    HCONTAINER hCon = NULL; //Container handle
    int uiSignType; //Signature type
    //------------------------------------------------------------------------
    //--------------------------2. Use certificate-------------------------------------
    //Use one of the two methods to obtain the certificate: 1. Enumerate all
```

This is an excerpt of the PDF (Some pages are marked off intentionally)

------ The End ------